

We claim:

1. A system comprising:

a plurality of inter-related first objects that share a predetermined password; and,
a plurality of first object-oriented interfaces defining methods supported by the inter-

5 related first objects, the first object-oriented interfaces publicly exposed by the inter-related first objects and queryable by second objects and the inter-related first objects to learn of the first object-oriented interfaces;

each first object-oriented interface including a password argument to limit access thereto to the inter-related first objects sharing the predetermined password.

10 2. The system of claim 1, further comprising a plurality of second object-oriented interfaces defining methods supported by the inter-related first objects, the second object-oriented interfaces publicly exposed by the inter-related first objects and queryable by the second objects and the inter-related first objects to learn of the second object-oriented interfaces.

15 3. The system of claim 2, wherein the second object-oriented interfaces are required by a predetermined specification, and lack password arguments to limit access thereto, the methods defined by the plurality of second object-oriented interfaces returning a type of "not implemented" message in response to invocation thereof.

4. The system of claim 3, wherein the second object-oriented interfaces correspond to the first object-oriented interfaces, such that each second object-oriented interface is a non-implemented and password-free version of one of the first object-oriented interfaces.

5. The system of claim 1, further comprising an object manager to manage the plurality of inter-related first objects and the second objects, the object manager responsive to requests from the inter-related first objects and the second objects to invoke the methods defined by the first object-oriented interfaces, and responsive to queries from the inter-related first objects and the second objects about the first object-oriented interfaces.

6. The system of claim 5, wherein the second objects at least partially represent one or more client applications running on the system.

7. The system of claim 6, wherein each inter-related first object represents a proxy between the object manager and system resources and acts as a client when invoking those of the methods supported by other of the inter-related first objects.

8. A system comprising:

a plurality of inter-related first objects that share a predetermined password;
a plurality of first object-oriented interfaces defining methods supported by the inter-related first objects, the first object-oriented interfaces publicly exposed by the inter-related first objects and queryable by second objects and the inter-related first objects to learn of the first object-oriented interfaces, each first object-oriented interface including a password argument to limit access thereto to the inter-related first objects; and,

a plurality of second object-oriented interfaces defining methods supported by the inter-related first objects, the second object-oriented interfaces publicly exposed by the inter-related first objects and queryable by the second objects and the inter-related first objects to learn of the second object-oriented interfaces,

5 wherein the second object-oriented interfaces are required by specification, and lack password arguments to limit access thereto, and wherein the methods defined by the plurality of second object-oriented interfaces return a type of "not implemented" message in response to invocation thereof, and

10 the second object-oriented interfaces corresponding to the first object-oriented interfaces, such that each second object-oriented interface is a non-implemented and password-free version of one of the first object-oriented interfaces.

9. The system of claim 8, further comprising an object manager to manage the plurality of inter-related first objects and the second objects, the object manager responsive to requests from the inter-related first objects and the second objects to invoke the methods
15 defined by the first object-oriented interfaces, and responsive to queries from the inter-related first objects and the second objects to invoke the methods defined by the second object-oriented interfaces.

10. The system of claim 9, further comprising a plurality of the second objects each second object at least partially representing a client application running on the system,
20 wherein each inter-related first object represents a proxy between the object manager and system resources and acts as a client when invoking those of the methods supported by other of the inter-related first objects.

11. A method comprising:

attempting to access a first object-oriented interface defining a method supported by an inter-related first object, by an inter-related second object, the first object-oriented interface including a password argument to limit access thereto, the first and the second
5 objects sharing a predetermined password, the first object-oriented interface having a corresponding second object-oriented interface defining a method supported by the first object, which is required by a predetermined specification and that lacks a password argument to limit access thereto; and,

passing the predetermined password as the password argument of the first object-
10 oriented interface by the second object such that the second object is able to invoke the method defined by the first object-oriented interface due to the predetermined password having been passed as the password argument of the first object-oriented interface.

12. The method of claim 11, further comprising attempting to access the first object-oriented interface by a third object not inter-related with the first and the second objects,
15 the third object not privy to the predetermined password, such that the third object is unable to invoke the method defined by the first object-oriented interface.

13. The method of claim 11, further comprising initially querying the first object to learn of the first object-oriented interface and the second object-oriented interface.

14. The method of claim 11, further comprising:

20 accessing the second object-oriented interface; and,

returning a type of “not implemented” message by the inter-related first object in response to access of the second object-oriented interface.

15. The method of claim 11, wherein the second object attempts to access the first object-oriented interface defining the method supported by the first object by requesting an object manager to invoke the method, such that the object manager acts as a broker between the first and the second objects.

16. An article of manufacture comprising:

a computer-readable medium; and,

means in the medium for implementing a first object-oriented interface and a second object-oriented interface both defining a method, the first object-oriented interface including a password argument to limit access thereto to inter-related objects, the second object-oriented interface lacking a password argument to limit access thereto.

17. The article of claim 16, wherein the means is further for returning a type of “not implemented” message in response to invocation of the method of the second object-oriented interface.

18. The article of claim 16, wherein the means is further for returning the first and the second object-oriented interfaces in response to queries therefor.

19. The article of claim 16, wherein the second object-oriented interface corresponds to the first object-oriented interface.

20. The article of claim 16, wherein the second object-oriented interface is required by specification.